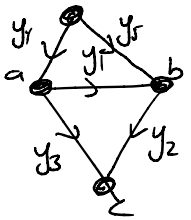
Applications

1. Kirchhoff's Current Law (recitation)



In an electrical network with nodes and edges, the current flowing into a node must equal the current flowing out of the node. This gives a system of equations (e.g. at node a, $y_4 = y_1 + y_3$)

This data can be encoded in an "incidence matrix"

The left nullspace of this matrix provides the solution set to this system

2. Hamming Codes

In digital communications, messages must be encrypted, accurately transmitted, then decrypted.

Sometimes there are errors in the transmission. To combat this, one uses error-correcting codes

In our model situation, we are transmitting bits (0 or 1). Moreover, we assume that errors are rare (so either 0 or 1 errors)

Alice sends message M to Bob

$M \xrightarrow[\text{encode}]{} C_A \xrightarrow[\text{(0 or 1 errors)}]{\text{send}} C_B \xrightarrow[\text{Correct}]{\text{Check \& Correct}} C_A \xrightarrow[\text{decode}]{} M$

How can we check and correct?

- Attempt 1: Send bits twice

	M		C_A
Alice encodes	0	as	00
and	1	as	11

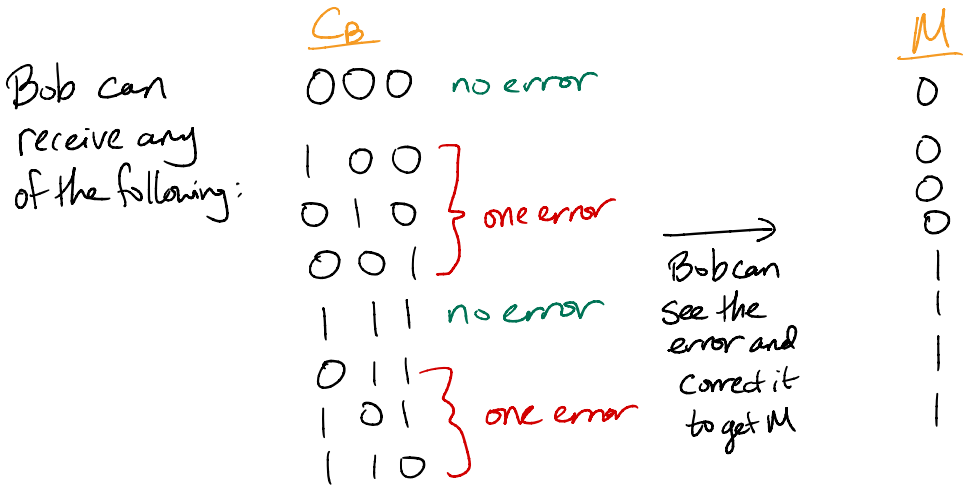
C_B	
00	no error
10	one error
01	one error
11	no error
01	one error
10	one error

Bob receives either

Bob knows that 01 and 10 contain an error, but he cannot correct them.

- Attempt 2: Send bits thrice.

	M		C_A
Alice encodes	0	as	000
and	1	as	111



The redundancy of an encoding scheme is:

$$\frac{\#\text{ of bits in } C_A}{\#\text{ of bits in } M}$$

The above scheme has redundancy 3.

The smaller the redundancy, the more efficient the scheme is. Hamming codes provide more efficient schemes.

The (7,4) Hamming code is defined as follows

Let $H = [F \mid I_3]$ where the columns of F contain all possible vectors with entries 0 & 1 and at least two 1's.

(H is the parity-check matrix)

Ex: $H = \left[\begin{array}{cccc|ccc} 1&1&1&0&1&0&0 \\ 1&1&0&1&0&1&0 \\ 1&0&1&1&0&0&1 \end{array}\right]$

F $\quad$ I_3

Key Fact: The matrix $\left[\begin{array}{c} I_4 \\ \hline -F \end{array}\right]$ has columns that form a basis of $N(H)$.

Ex: $\left[\begin{array}{c} I_4 \\ \hline -F \end{array}\right] = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ -1&-1&-1&0 \\ -1&-1&0&-1 \\ -1&0&-1&-1 \end{bmatrix}$

$a_1\ a_2\ a_3\ a_4$

Note $Ha_i = \vec{0}$ for each i. (check)

Now since $\dim N(H) = 4$ (check)
and a_1, a_2, a_3, a_4 are lincarly indep (check)
they form a basis.

Now, let $G = \left[\begin{array}{c} I_4 \\ \hline F \end{array}\right]$ (G is called the generating matrix)
We replace the -1's with 1's because we are only dealing with bits (0's and 1's).

Ex: $G = \left[\begin{array}{c} I_4 \\ \hline F \end{array}\right] = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&1&1&0 \\ 1&1&0&1 \\ 1&0&1&1 \end{bmatrix}$

From now on, all calculations are conducted (mod 2)

That is: Odd #s become 1
Even #s become 0

Since $G = \begin{bmatrix} I_4 \\ \hline -F \end{bmatrix}$ (mod 2), the columns of G form a basis for $N(H)$ (mod 2)

The process:

A message M is a vector $M \in \mathbb{R}^4$ with entries 0 or 1.
Now encode M by: $C_A = GM \in \mathbb{R}^7$ (mod 2)

Ex: $M = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \Longrightarrow C_A = GM = \left[\begin{array}{c} 1 \\ 0 \\ 1 \\ 1 \\ \hline 0 \\ 1 \\ 0 \end{array}\right]$ (M / parity check bits)

Key: C_A is a L.C. of the basis vectors of $N(H)$ (mod 2)
Hence $C_A \in N(H)$.

Now Alice sends C_A to Bob who receives C_B.

Error detection: $C_A = C_B \iff C_B \in N(H) \iff HC_B = \vec{0}$ (mod 2)

Error correction: Since there can only be one error, if there is an error, then $C_B = C_A + e_i$ (mod 2) for some $1 \le i \le 7$
↑ the i^{th} bit is wrong

Which i? Well:

$$HC_B = H(C_A + e_i) = HC_A + He_i = He_i = i^{th} \text{ column of } H \pmod 2$$

So if $HC_B = i^{th}$ column of H, then $C_A = C_B + e_i \pmod 2$

↑ change i^{th} bit of C_B to get C_A

Ex:

- Bob receives $C_B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

  Then $HC_B = \vec{0}$ (Check) $\Rightarrow C_A = C_B \Rightarrow M = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

- Bob receives $C_B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$

  Then $HC_B = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} =$ 3rd column of H

  $$\Rightarrow C_A = C_B + e_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \} M$$

  $$\Rightarrow M = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Why use this? The $(7,4)$-Hamming code has redundancy $\frac{\#\text{ bits of } C_A}{\#\text{ bits of } M} = \frac{7}{4} < 2$ (not bad at all)

General $(2^m-1, 2^m-m-1)$ Hamming Code

- $H = \left[\begin{array}{c|c} F & I_m \end{array}\right]$ where F is an $m \times (2^m-m-1)$ containing all possible nonzero vectors containing 0's & 1's and at least 2 1's.

- $G = \left[\begin{array}{c} I_{2^m-m-1} \\ \hline F \end{array}\right]$

The process is identical to the (7,4) Hamming Code.

Redundancy is $\frac{2^m-1}{2^m-m-1} \rightarrow 1$ as m gets large.